MCD Magazine

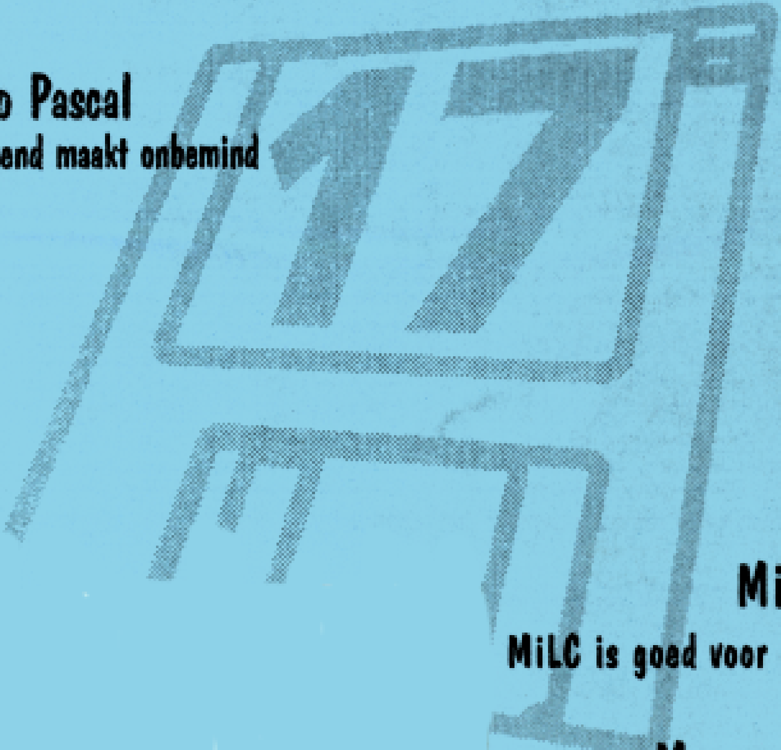
Het clubblad van MSX Club Drechtsteden

MSX2 Emulator

Nog niet af, maar het wordt heel mooi

Turbo Pascal

Onbekend maakt onbemind



MiLC

MiLC is goed voor elk!

Moonsound

De ultime geluidsuitbreiding voor de MSX

Jaargang 4 | Nummer 17 | September/Oktober 1994

COLOFON

Voorzitter
Hans Meijers, 078-511156

Penningmeester
Wim de Moor

Secretaris
Arjan Steenbergen, 078-146378

Redactie
Arjan Steenbergen, 078-146378
Mark Sletterink, 078-167682

Bestuursleden
Yvonne Schuit, 078-160023
Henk Groffen, 078-184714

Vormgeving
Arjan Steenbergen

Correspondentieadres
MSX Club Drechtsteden
Slangenburg 138
3328 DR Dordrecht
Tel. 078-511156

Donatie & Contributie
Giro 724696
t.n.v. MSX Club Drechtsteden
Melijn 12, 4761 NN
Zevenbergen

Inzenden materiaal
Stuur inzendingen naar:
Arjan Steenbergen
Mauritsweg 11
3314 JG Dordrecht
(per modem is ook mogelijk)
Fidonet: 2:285/36.6
Internet: arjann@dds.hacktic.nl

Medewerkers aan dit magazine
Bas Vijfwinkel
Sandy Brand

Copyright (C) 1994
Niets uit deze uitgave mag vermenigvuldigd en/of gepubliceerd worden zonder bronvermelding.

Inhoud

Voorwoord 2
MSX2-Emulator 3
Turbo Pascal 6
MoonSound 10
Listing Animatie 15
MiLC 20

Agenda

September

2	clubavond
16	clubavond
17	Beurs in Zandvoort
30	clubavond

Oktober

1	Beurs in Almelo
14	clubavond
28	clubavond

November

11	clubavond
25	clubavond

December

9	clubavond
23	clubavond

Voorwoord

Beste MSX vrienden,

De laatste tijd zit ik vaak te denken wat de toekomst ons zal brengen. Het aantal mensen die de MSX trouw zijn en blijven neemt af. Of men stopt helemaal met computers of men gaat over op een ander systeem. Op onze OpenDag in Juni was de aanloop wel goed maar er is weinig zaken gedaan. Ik hoop dat de standhouders toch genoeg redenen zien om volgend jaar weer te komen. Toch gaat het met de club goed. De avonden worden goed bezocht. Op de avonden is het een gezellige drukte. Het aandeel vrouwelijke leden zou nog wel hoger kunnen zijn. Ook zou ik een oproep willen doen aan leden die met een bepaald programma heel goed overweg kunnen. Zij zouden misschien een avond kunnen verzorgen over het betreffende programma (NvdR., of er een artikel over schrijven voor in MCD Magazine). We kunnen nog wel wat ideeën gebruik-en dus spui ze maar bij ons! Na de vakantie hoop ik iedereen weer gezond en wel tegen te komen met veel enthousiasme voor de club en misschien wel met wat leuke ideeën. Groetjes,

Hans Meijers
(voorzitter MCD)

Verhuisbericht

Onze penningmeester, Wim de Moor, is sinds 3 maanden verhuist van de Jacob van Lennepstraat in Dordrecht naar Melijn 12, 4761 NN te Zevenbergen. We wensen hem en zijn gezin veel plezier in hun nieuwe woning.

All wyht. Rho snitched mg kegtops awound?

MSX2-Emulator

In dit berichtje van Albert Siersema, afkomstig van Fidonet, vindt de u de laatste informatie over de MSX2-emulator.

NvdR. het originele bericht op Fidonet was in het engels geschreven. Ik heb daarom het bericht zo letterlijk mogelijk vertaald in het Nederlands zodat iedereen kan meegenieten.

Voor de nieuwsgierigen. Ik heb even HEEL snel iets in elkaar gevogeld. Beschouw het niet volledig als waarheid of voorspelling.

Er zijn nu zoveel vragen gekomen dat dit lijstje me wel nuttig lijkt. Er staat tevens in waarom we zo lang bezig zijn :-)

<u>De MSX-2 emulator voor MS-Dom</u>
Informatie onder voorbehoud!
vertaling van engelse versie

De MSX-2 emulator is volledig overnieuw geschreven, zodat alle code nu door onszelf geschreven is. Hij is nog steeds volledig in 80386 assemblercode geschreven. Hij omvat nu al 1Mb aan (T)ASM listings.

Systeemeisen

- 80386 (DX-40Mhz voor echte MSX video snelheid)
- EMS geheugen, ongeveer 1Mb
- Ongeveer 550Kb (vrij) conventioneel geheugen (sorry, maar hebben veel buffers nodig voor snelheid e.d. Er is helaas niet veel ruimte over voor drivers.)
- VGA

- PC-Speaker of AdLib compatible voor het geluid
- Muis met driver voor MSX-muis emulatie

Nu volgen in het kort wat uitbreidingen die we al hebben toegevoegd of die we van plan zijn toe te voegen:

- MSX-2
- Compleet herschreven Z-80 emulatie. Zelfs onofficiële flags.
- Memory mapper. Op dit moment 128Kb of 256Kb, maar dat zal instelbaar worden in (waarschijnlijk) stappen van 64Kb.
- Geheugen indeling van Philips MSX-2 machines (slot 3 expanded). Misschien voegen we verschillende layouts toe voor hardware specifieke programma's
- Instelbare keyboard indeling
- Instelbare drive indeling. Ook een 1 drive MSX met phantom drive ondersteuning en CTRL bij het opstarten.
- De disk routines werken nu echt goed, bijv. MSX-DOS werkt nu en FORMAT ook.
- Memory protection is nu véél beter.
- Muis ondersteuning (hoewel we nog steeds problemen hebben met sommige programma's (zoals EASE) die de muis op een andere manier lijken te lezen. Alle programma's die de BIOS gebruiken of die hetzelfde algoritme gebruiken om de muis te lezen werken wel.
- Alle MSX-2 video modes. Nu geheel in VGA (realistische kleuren).
- PSG emulatie (AY-3-89-10)
- Philips muziekmodule emulatie
- Konami SCC emulatie
- Weergave van muziek en geluiden door de PC-speaker, Adlib of compatible en misschien GUS en/of Soundblaster ondersteuning.
- MSX-2 Clockchip emulatie

Het volgende moeten we nog allemaal doen:

- Z80 emulatie is bijna perfect, maar een programma genaamd TESTCPU (Z80A test programma) werkt nog steeds niet 100% :-(
- 100% MSX snelheid in plaats van alleen topsnelheid
- MSX-2 video (sprites, MSX-2 video modes, pfff... veel experimenteren)
- Joystick
- Printer
- Cassette

- Volledige MSX-Audio / MSX-Music ondersteuning (maar we hebben nog steeds te weinig info)
- Muis routines 100% compatible
- Wat kleine details in de disk routines
- Configuratie programma
- Ondersteunende utilities
- Documentatie

We zijn nu terug van onze vakantie en we zijn al weer druk bezig met programmeren. We kunnen echter nog steeds veel documentatie gebruiken...

The "CompuJunks": Albert Siersema, Ronald Gorter, Peter Kelders

FidoNet

Albert Siersema op 2:282/517

InterNet

Albert.Siersema@prgbbs.idn.nl (het addres in de MSX-1 emu docs werkt nog wel maar het is mogelijk dat dit adres (appie@hiogron.rhg.nl) ongeldig zal worden in de nabije toekomst, terwijl prgbbs.idn.nl altijd goed is.

BBS

Filosoft/Programmers BBS
Tel. 050-412288 of 050-426071

MCD PC-utilities

Diverse Public Domain utilities voor de PC om het combineren van de PC en de MSX te vergemakkelijken

* **MSX1-Emulator 0.3b**
  de alom bekende emulator van 'de compujunks'.
* **MSX Graphics Converter 1.00**
  converteert MSX-plaatjes naar het PCX-formaat
* **MSX Text Converter 1.00**
  converteert MSX-teksten naar de PC en andersom
* **Graphic Workshop 7.0c**
  handige convert utilitie voor de PC

Verkrijgbaar vanaf de beurs in Zandvoort

Turbo Pascal

Deze programmeertaal van Borland is eigenlijk een wat ondergewaardeerde taal op de MSX. Laten we daar maar eens wat aan gaan doen.

De programmeertaal Pascal is in 1969 ontworpen door professor Niklaus Wirth, De taal is genoemd naar de Franse wiskundige Blaise Pascal (1623-1662), die reeds in 1640 een rekenmachine uitvond. De taal is enorm populair geworden, mede doordat het maken van een compiler relatief eenvoudig was. Wirth schreef zelf de eerste compiler en wel in Pascal zelf. In 1980 ontstond de behoefte de taal officieel te standaardiseren en enkele dubbelzinnigheden en onduidelijkheden te veranderen. In 1983 is de definitieve standaard vastgelegd in het document ISO 7185. In de loop der tijd zijn er veel pascal-compilers gemaakt, en veel fabrikanten hebben de taal uitgebreid of ingeperkt, zodat Pascal-programma's lang niet altijd van de ene naar de andere computer overdraagbaar zijn.

De meest bekende Pascal-compiler is zonder twijfel Turbo Pascal van Borland. Borland heeft een Pascal-compiler gemaakt die de volledige ISO standaard omvat maar daarbij ook een heleboel uitbreidingen bevat. In 198? heeft Borland Turbo Pascal ook voor de MSX uitgegeven. Oorspronkelijk was deze compiler voor CP/M geschreven, maar door wat aanpassingen hier en daar kon Borland ook een MSX versie uitbrengen. Of er ook andere versies zijn uitgegegeven zijn weet ik niet maar versie 3.00 voor de MSX is compatible (gelijkwaardig) met versie 3.00 op de PC. Er slechts enkele kleine verschillen voor wat

betreft de schermaansturing en het gebruik van grafische modes. Het grafische modes is in de MSX versie namelijk niet mogelijk.

Als u Turbo Pascal nog niet bezit, of u zou de originele handboeken er ook wel eens bij willen hebben, dan krijgt u daarvoor de kans op de beurs in Zandvoort. Turbo Pascal van Borland zal dan officieel verkocht worden door MSX Club Enschede. Ik heb ook vernomen dat er plannen zijn om GIOS, een grote verzameling handige routines, in Turbo Pascal in te bouwen. Dit zou natuurlijk fantastisch zijn. Zo kunt u grotere programma's maken (doordat veel routines ergens in de mapper hangen) en het scheelt tijd bij het compileren.

Indien u van plan bent om in Turbo Pascal te gaan programmeren dan kan ik u het boekje 'Turbo Pascal binnen handbereik' aanraden. Hierin staan alle Pascal commando's beschreven en uitgelegd. Het boekje kost slechts f15,=. dit boekje is natuurlijk gebaseerd op de PC versie, maar bij ieder commando staat vanaf welke versie het commando bestaat.

Installeren

Als u nog niet eerder met Turbo Pascal gewerkt heeft is het verstandig om nu eerst het programma TINST te starten, en voor veel gebruikte commando's handige toetsen te kiezen.

Het gebruik

Wanneer we Turbo Pascal opstarten wordt er eerst gevraagd of we de foutmeldingen in willen laden. Kies hier voor Yes, anders krijgen we bij een fout alleen maar het foutnummer te zien in plaats van een omschrijving. Dit is vooral in het begin erg onduidelijk. Na deze eerste vraag krijgen we het hoofdmenu te zien. Dit ziet er ongeveer uit zoals in figuur 1 (op de volgende bladzijde).

We kunnen de commando's gebruiken door de hoofdletter in typen, dus de E van Edit, de C van Compile en de X van eXecute. Laten we alle opties eens nalopen.

Logged drive- Dit is de drive waar u op werkt.
Work file - Dit is de file die u op dit moment aan het editten bent.
Main file - Deze file wordt

```
Logged drive: A

Work file:
Main file:

Edit      Compile  Run   Save

eXecute   Dir      Quit  compiler Options

Text:      9 bytes (0110-0118)
Free: 20717 bytes (0119-B206)

>█
```

■Figuur 1

gecompileerd als u voor Compile kiest in het hoofdmenu. Als u programma uit meerdere bestanden bestaat en u nu met een deel daarvan bezig bent, dan kunt u met Main file opgeven welke file gecompileerd moet worden. Dan hoeft u niet steeds het hoofddeel van uw programma op te halen als u wilt testen of een stuk programma werkt.

Edit - dit spreekt natuurlijk vanzelf, het editten van de programmacode in de editor

Compile - dit lijkt me logisch, het compileren van een Pascal programma naar disk of geheugen.

Run - met Run kunnen we het succesvol gecompileerde programma starten en testen.

Save - het bewaren van het programma

eXecute - hiermee kunnen we een losstaand programma opstarten, bijvoorbeeld TED of iets dergelijks.

Dir - hetzelfde als het gelijknamige MSX-DOS-commando

Quit - de Turbo Pascal omgeving verlaten.

compiler Options - Als we voor deze optie kiezen krijgen we een tweede menu te zien. Dit ziet er uit als in figuur 2 (onderaan de pagina). We kunnen nu kiezen of we willen compileren naar het geheugen, naar een COM-file of naar een CHN-file (chain-file). Als we voor compileren naar een com- of chain-file kiezen kunnen we ook nog begin- en eindadressen opgeven. Met 'Find run-time error' kunnen we opsporen waar in het programma een fout is opgetreden tijdens de uitvoering daarvan. Dit zijn zogenaamde run-time errors. Als tijdens het runnen van een programma een fout optreedt dan krijgt u een melding te zien in de trant van:

```
compile -> Memory
           Com-file
           cHn-file

Find run-time error  Quit

>█
```

■Figuur 2

```
run-time error xx, PC=002D
Program aborted
```

Als we het adres wat met PC wordt aangeduid opgegeven bij 'Find runtime error' dan kunnen we kijken waar in het programma de fout is opgetreden. Met Quit keren we weer terug naar het hoofdmenu.

Zo, nu we weten hoe de compiler werkt kunnen we aan de slag gaan. We beginnen even opnieuw en starten Turbo Pascal op en kiezen Yes bij de eerste vraag. We gaan een nieuw programma maken dus we kiezen de W van 'Work file'. Er wordt nu gevraagd op een Work file name. Vul hier een bestandsnaam in, bijvoorbeeld test. De extensie .PAS wordt automagisch toegevoegd. Er wordt nu gezocht of het bestand TEST.PAS al bestaat. Als dat niet het geval is komt er 'New file' te staan. We gaan nu een klein programmaatje intypen. Kies voor de E van 'Edit'. U krijgt nu een leeg scherm te zien, met aan de bovenkant een statusregel. Typ nu het volgende programmaatje maar eens in:

```
PROGRAM test;
BEGIN
  writeln ('hello!');
END.
```

Als u dit heeft ingetypt, druk dan op CTRL+C of op CTRL+STOP om de editor te verlaten. U krijgt dan onderaan het scherm weer de prompt te zien (>). Als u nu op een willekeurige toets (bijvoorbeeld de spatiebalk) drukt, dan krijgt u het hoofdmenu weer te zien. Kies nu voor de C van compile. Als alles goed gaat krijgt u het volgende te zien als in figuur 3 (onderaan de pagina). Als eventuele de fouten hersteld heeft en het compileren is goed gegaan dan gaan we het programma starten. Druk daarvoor op de R van Run. Als alles goed gegaan is ziet u nu de tekst 'hello!' verschijnen. Later zal ik nog wel uitleggen wat het programmaatje precies doet. In het volgende magazine komt er eerst wat algemene Pascal theorie en daarna gaan we echt programmeren.

Arjan Steenbergen

```
Compiling
  5 lines

Code:     53 bytes (8155-818A)
Free:  21619 bytes (818B-D5FE)
Data:      7 bytes (D5FF-D606)
```

■Figuur 3

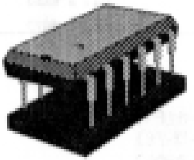
MoonSound

Er wordt hard gewerkt aan een nieuwe geluids-uitbreiding voor de MSX: een cartridge met een OPL4-chip erop!

Al bijna anderhalf jaar geleden werd ik door het Zwitserse MSX-Händlergemeinschaft benaderd met het verzoek of ik de software zou willen schrijven voor een nieuwe soundchip die ze voor de MSX wilden gaan gebruiken. Het plan was om een cartridge te maken voor MSX met daarin de Yamaha OPL3. Het hele project nam nogal wat tijd in beslag zodat Yamaha ondertussen de opvolger al uit bracht: de OPL4! Na het lezen van de OPL4 specificaties heb ik er op aangedrongen om de OPL4 te gaan gebruiken en gelukkig zijn ze hiermee akkoord gegaan. De naam van deze nieuwe OPL4-cartridge is Moonsound. Helaas heb ik op het moment van schrijven nog steeds geen OPL4 evaluation board. Wel heb ik gelukkig al ruim een half jaar een OPL3 evaluation board. Dat betekent dat alles in dit artikel met betrekking tot de FM synthesizer ook echt getest is, maar alles mmet betrekking tot de PCM synthesizer uit de Yamaha manuals komt zonder dit te testen.

Deze tekst is afkomstig uit InfoDrome BBS (01859-13957)

Om te beginnen zal ik eerst een opsomming geven van de belangrijkste mogelijkheden van de chip. Ik zal proberen niet al te technisch te worden, maar dit is soms niet helemaal te vermijden. Enige ervaring met het gebruik van de MSX-Audio (Music Module) en MoonBlaster zal al veel verhelderen.



De OPL4 heeft 2 hoofdonderdelen Een FM Synthesizer Unit (de OPL3) en een PCM Synthesizer Unit. Deze laatste staat in de PC wereld ondertussen beter bekend als Wave Table Synthesis.

FM

- Maximaal achttien 2-Operator FM kanalen of vijftien FM kanalen met daarnaast vijf drumgeluiden
- Maximaal zes 4-Operator FM kanalen (met daarnaast nog zes 2-Operator kanalen)
- Acht verschillende waveforms
- Mogelijkheid om per kanaal aan te geven of deze links, rechts of zowel links als recht te horen is

PCM:

- Maximaal 24 waves tegelijk! (24 kanalen dus)
- Afspeelfrequentie van 44.1 kHz
- 8, 12 of 16 bits wavedata
- Stereo signaal in 16 standen voor elk kanaal

De FM Synthesizer

Om de FM synthesizer van de OPL4 te kunnen beoordelen kunnen we deze het beste vergelijken met de MSX-Music (FM-PAC) en de MSX-Audio (Music Module). De mogelijkheden van deze beide chips zullen iedereen ondertussen wel bekend zijn. De OPL4 lijkt het meeste op de MSX-Audio. Dit is ook logisch omdat de MSX-Audio eigenlijk de OPL1 is (NvdR. OPL2 toch? hij heeft 2 operators). De OPL4 is 100% compatible met de MSX-Audio waar het de FM Synthesizer betreft. Bij de OPL4 zijn echter twee keer zoveel kanalen mogelijk!

Bovendien zijn er de zogenaamde 4-Operator instrumenten. Normaal gesproken wordt een instrument gemaakt door twee operatoren op de juiste manier in te stellen. Denk maar aan het Make Own Voice deel in MoonBlaster. Nu is er echter de mogelijkheid om ook instrumenten te maken met vier operatoren. Het spreekt natuurlijk voor zich dat hiermee (met enige moeite) mooiere instrumenten gemaakt kunnen worden dan tot nu toe gebruikelijk is. Een andere verbetering bij het maken van instrumenten is dat nu per operator het type waveform ingesteld kan worden. Zonder hier al te diep op in te gaan even dit: bij de MSX-Audio is de waveform altijd een sinus, deze kan nu dus gewijzigd worden.

Tenslotte is er aan de FM-synthesizer een stereo mogelijkheid toegevoegd. Moest er in MoonBlaster nog veel moeite gedaan worden met twee verschillende chips om dit te verkrijgen, nu kan er gewoon per kanaal ingesteld worden of deze links, rechts of links en rechts te horen is.

De PCM Synthesizer

Dit onderdeel van de OPL4 maakt het mogelijk om samples af te spelen. Een sample is een stukje gedigitaliseerd geluid. Dit zorgt er dus voor dat een instrument heel realistisch weergegeven kan worden, namelijk een exacte kopie! Nadeel van een sample is dat een grote afwijking in opname en afspeelfrequentie zorgt voor een lelijk resultaat: het instrument lijkt niet meer op het origineel. Om bijvoorbeeld een mooie piano te kunnen creëren zou dus bijna elke noot van de piano apart gedigitaliseerd moeten worden.Om dit te voorkomen wordt het gedigitaliseerde geluid beïnvloed door de PCM synthesizer. Hier kunnen ook nog allerlei extra zaken opgegeven worden. De meeste hiervan lijken op die van FM synthese en hebben te maken met de aanzwelsnelheid e.d. van het geluid. Bijkomend voordeel van deze synthesizer is dat hierdoor de samples vaak kort gehouden kunnen worden omdat bijvoorbeeld het uitsterven van een toon door de synthesizer geregeld kan worden. Het afspelen van een sample met deze extra 'effecten' is nu het bekende Wave Table Synthesis en een sample met die extra informatie wordt (hier) een wave genoemd. De kwaliteit van een sample wordt door twee dingen bepaald: de sample resolutie en de sample snelheid. Om hier diep op in te gaan voert hier wat te ver, maar met een sample resolutie van 16 bits en een frequentie van 44,1 kHz heeft de OPL4 geluid van CD kwaliteit!

Ook de PCM synthesizer heeft stereo mogelijkheden. Deze zijn nog beter dan die van de FM synthesizer, want hier kan in 16 stappen worden bepaald hoe hard het geluid links en rechts moet klinken. Dit is dus geen scheiding tussen links of rechts, maar een echte balansregeling! Voor elk van de 24 kanalen kan dit ingesteld worden.

PCM data neemt veel geheugenruimte in, zelfs met de synthe-

sizertruukjes van de OPL4. Zeker op MSX is het lastig als voor een liedje zo'n 300 kB aan wavedata ingeladen moet worden. De oplossing hiervoor is de 2MB ROM die ook in de cartridge komt te zitten. Dit is een ROM die door Yamaha speciaal is ontworpen voor de OPL4 met daarin alle General Midi geluiden! Dit zijn 128 instrumenten met daarnaast nog eens 47 drumgeluiden. Bovendien wordt er nog een 128 kB RAM bij gezet om er voor te zorgen dat ook eigen samples gebruikt kunnen worden. Voor de echte freaks is er nog de mogelijkheid deze 128 kB RAM te vervangen door 512 kB RAM.

Voor alle duidelijkheid: de OPL4 kan dus geen geluid digitaliseren, alleen gedigitaliseerd geluid afspelen. Dit is echter niet zo'n probleem aangezien de wavedata niets anders is dan kale PCM data met wat extra informatie. Het zou dus eenvoudig moeten zijn om samples te gebruiken die op een Music Module, Turbo-R of een willekeurige andere computer gemaakt zijn.

Snelheid

Voor programmeurs kan het interessant zijn om te weten dat de OPL4 veel sneller is dan de MSX-Music en MSX-Audio. Bij deze laatste twee moest met een toch trage Z80 soms nog gewacht worden op de chips. De OPL4 is zo snel dat zelfs de R800 niet meer hoeft te wachten. Ander handigheidje is dat alle(!) registers nu ook kunnen worden gelezen. Voor de relatief trage MSX computer is het een enorm voordeel dat de OPL4 vrijwel alles zelf afhandelt. Als dit niet het geval was zou het nooit mogelijk zijn om 24 waves tegelijk af te spelen. Denk maar aan de toch knap geprogrammeerde MODplayer van Xela-Soft. Deze heeft op een Turbo R al moeite met het spelen van 4 samples tegelijk. De OPL4 maakt het mogelijk demo's of spellen te maken met geweldige achtergrondmuziek. Bij spellen kan de 128 kB RAM perfect gebruikt worden om samples als

geluidseffecten te gebruiken. Het zou ook mogelijk moeten zijn om een MODplayer te maken die op de Z80 werkt en nog genoeg tijd over te houden!

De software

Bij de aankoop van de OPL4 zullen er twee muziekprogramma's bijgeleverd worden. Dit zijn allebei speciale MoonBlaster versies. Er is een FM versie en een Wave versie. De FM versie heeft de mogelijkheid om (maximaal) 18 FM kanalen te gebruiken in combinatie met 6 Wave kanalen voor bijvoorbeeld de drums. De Wave versie zal 'alleen' de 24 Wave kanalen ondersteunen. Dit onderscheid is gemaakt om de songdata niet al te groot te maken en om de programma's op computers met 128 kB RAM te laten werken. Zoals gezegd zullen beide programma's erg op MoonBlaster lijken en iedereen die gewend is hiermee te werken zal dan ook snel aan deze programma's gewend zijn. Het grootste verschil met MoonBlaster is dat deze programma's onder DOS zullen werken. Indien er een 256kB memory mapper aanwezig is zal ook DOS2 (en dus harddisk) ondersteund worden. In beide programma's zal het mogelijk zijn om MoonBlaster v1.4 muziek in te laden.

Verder zal ik zodra ik deze twee programma's af heb beginnen met het programmeren van een General MIDI file player en een Amiga MODplayer. Wanneer deze af zijn durf ik niet te zeggen, maar wanneer alles zo eenvoudig is als het volgens de Yamaha specifica-ties zou moeten zijn zal dit niet al te lang duren. Vele groepen hebben al aangekondigd dat ze de OPL4 gaan ondersteunen, op de beurs in Tilburg zullen al muziekdemo's en spellen worden uitgebracht met OPL4 muziek.

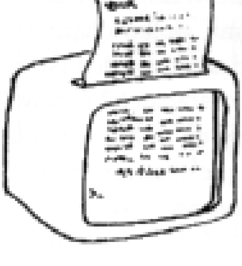
Moonsound (de cartridge voor MSX met OPL4) zal waarschijnlijk in oktober of november leverbaar zijn, de prijs zal rond de *f*350,- liggen. Op de beurs in Zandvoort zullen de exacte levertijd en prijs bekend worden gemaakt. Ga voor een demonstratie van de OPL4 langs bij de stand van Sunrise. Moonsound kan daar ook worden besteld.

Remco Schrijvers
Moonsoft /Sunrise

Listing Animatie

Mr. Retaliator geeft ons tips over het aanbrengen van duidelijke markeringen in listings

Heeft u dat ook wel eens? In uw BASIC listing van 20 KB loopt u zich steeds rot te zoeken naar een bepaalde routine. Je kunt met REM regels wel afzonderlijke delen markeren, maar dat neemt een hoop ruimte in beslag (dit is met name een groot nadeel als je de KUN-compiler gebruikt). Met deze ML-routine kunt u echter één character een speciale functie toewijzen, namelijk animatie. “Animatie?”, zul je je nu wel afvragen. Voor degene die dit nog niet wisten: de tekens (of characters) op het MSX scherm kunnen gewoon veranderd worden. Het feit dat uw lettertjes er altijd hetzelfde uit zien komt door de BIOS; tijdens het opstarten van uw MSX haalt deze de tekenset ergens uit het ROM en zet die in het VDP geheugen neer. Tijdens het schrijven van de routines stuite ik echter op onverwachte problemen die ik anderen wil besparen. Eerst wilde ik gewoon elke interrupt de patroon tabel verranderen met behulp van ‘direct access’ (bijvoorbeeld met behulp van LDIRVM). Dit gaat echter niet. De BASIC interpreter en de BIOS gebruiken deze ‘direct access’ VDP registers dan al voor hun eigen doeleinden, en als je deze registers elke interrupt gaat aanpassen is de kans groot dat het hele beeld een inhoud krijgt die voor ons simpele wezens weinig betekenis heeft (oftewel: puin!). Dan maar ‘indirect’ proberen (met het PSET commando dus), dit gaat in grafische schermen wel goed maar bleek in text schermen niet helemaal naar behoren te func-



tioneren (logisch!). Na even nagedacht te hebben kwam ik op het volgende idee: het VRAM blijft 'onbeschadigd' als ik een andere scherm modus (bijv. SCREEN 8) instel in de VDP (niet zoals in BASIC want dan wist de interpreter het scherm). Door een paar good gemikte PSETjes moest het dan mogelijk zijn om de patroon tabel te wijzigen. Het bleek echter dat VRAM adres 1 in screen 0 overeen komt met VRAM adres 2 in SCREEN 8. Dat betekend dus dat ik steeds een pixel in SCREEN 8 moet overslaan als ik een serie achter elkaarstaande bytes in screen 0 wil verranderen. Nadat ik dit probleem verholpen had kwam er nog een leuk probleem: steeds als de VDP werdt ingesteld naar SCREEN 8 ontstond er onder (!) de functietoetsen een gekleurde balk. Na lang zoeken bleek dat dit de 'border' van het scherm was die van kleur verranderde. Het 'screen display disabled' maken hielp niet. Nog meer zoekwerk bracht een dader aan het licht: register 7 (zie figuur 1). In SCREEN 0 staat hier de voorgrond en achtergrond (=border) kleur van de characters in. Maar omdat in SCREEN 0 met een instelbaar kleuren palet wordt gewerkt staat hier het kleurnummer in terwijl in SCREEN 8 in dit register de directe palet instelling staat (zie fig.2). Doordat er van scherm wordt gewisseld verranderd dus de betekenis van register 7. Ik heb dit voor de eenvoud maar opgelost door COLOR 15,0,0 als default kleurinstelling in scherm 0 te nemen en een 0 in register 7 te zetten als de VDP in screen 8 staat (normaal is de RGB waarde van kleur 0: 0,0,0; dit is dus hetzelfde als die van kleur nummer 0 in screen 8). Toen bleek alles naar behoren te werken, eindelijk!

Gebruik

Kleuren: `color 15,0,0`
`&HA000` - avctiveer routine
`&HA002` - deactiveer routine.
`&HA004` - X
`&HA005` - Y.
`&HA006` - animatiesnelheid.
deze waarde moet altijd >0 zijn!

Het bijgeplaatste basic listinkje berekend die waardes X en Y, ze geven aan welke character geanimeerd moet worden (standaard is dit character 62 (oftewel ">") en gaat de routine uit van width 80. Veel plezier er mee. Dangaioh beam!

Retaliator

BASIC Programma:

```
10 PRINT "In screen 0, width 40 is ";
20 PRINT "startadres patroontabel &H0800"
30 PRINT "In screen 0, width 80 is ";
40 PRINT "startadres patroontabel &H1000"
45 PRINT
50 INPUT "Startadres patroontabel:";B
60 INPUT "Character number:";Q
70 W=(B+(Q*8))*2:Y=INT(W/256):X=W-(Y*256)
80 PRINT"X:";X,"Y:";Y
```

Figuur 1:

Inhoud in SCREEN 0:

	MSB 7	6	5	4	3	2	1	0 LSB
R#7:	TC3	TC2	TC1	TC0	BD3	BD2	BD1	BD0

TC3 tot TC0 = Text Color (Tekst Kleur)
BD3 tot BD0 = Back Drop color (Achtergrond Kleur)

Figuur 2:

Inhoud in SCREEN 8:

	MSB 7	6	5	4	3	2	1	0 LSB
R#7:	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0

BD7 tot BD0 = Back Drop color (Achtergrond Kleur)
Kleur codering in SCREEN 8: het kleurnummer is een verzameling bitjes die een RGB codering voorstellen:

	MSB 7	6	5	4	3	2	1	0 LSB
R#7:	G2	G1	G0	R2	R1	R0	B1	B0

G2 tot G0 = intensiteit groen (0 - 7)
R2 tot R0 = intensiteit rood (0 - 7)
B1 en B0 = intensiteit blauw (0 - 3)
In BASIC kun je de kleurcode als volgt berekenen:
KLEUR = R*4 + G*32 + INT(B/2)

CHARANIM.ASM

```
        ORG     &HA000
H_TIMI: EQU     &HFD9F
VDP0AD: EQU     &HF3DF
NRPATR: EQU     3

        JR      B_HOOK
        JR      R_HOOK

;data field
XY_POS: DEFB    224
        DEFB    35
WAITLN: DEFB    4

;remove hook
R_HOOK: DI
        LD      HL,O_TIMI
        LD      DE,H_TIMI
        LD      BC,5
        LDIR
        EI
        RET

;build hook +
;search for VDP ports
B_HOOK: LD      HL,PORT1R
        LD      A,(6)
        INC     A
        LD      (HL),A
        INC     HL
        LD      A,(7)
        INC     A
        LD      (HL),A
        INC     HL
        INC     A
        INC     A
        LD      (HL),A
        DI
        LD      HL,H_TIMI
        LD      DE,O_TIMI
        LD      BC,5
        LDIR
        LD      HL,N_TIMI
        LD      DE,H_TIMI
        LD      C,5
        LDIR
        EI
        RET

;actual code
CODE:   DI
        LD A,(COUNTR)
        DEC A
        JR NZ,JUMP06
        LD A,(WAITLN)
JUMP06: LD (COUNTR),A
        JP NZ,O_TIMI
;INIT +
;set VDP to SCREEN 8
        LD      A,(PORT1W)
        LD      C,A
        LD      A,2
        OUT     (C),A
        LD      A,15+128
        OUT     (C),A
        XOR     A
        OUT     (C),A
        LD      A,7+128
        OUT     (C),A
        LD      A,(VDP0AD)
        OR      &B00001110
        OUT     (C),A
        LD      A,0+128
        OUT     (C),A
        LD  A,(VDP0AD+1)
        AND     &B11100111
        OUT     (C),A
        LD      A,1+128
        OUT     (C),A
;copy new patterns
;to VRAM
        CALL    WAITDP
        LD      A,36
        OUT     (C),A
        LD      A,17+128
        OUT     (C),A
        LD      D,C
```

```
        LD      A,(PORT3W)
        LD      C,A
        LD      B,10
        XOR     A
JUMP01: OUT     (C),A
        DJNZ    JUMP01
        LD      C,D
        LD      B,16
        LD      HL,(XY_POS)
        EX      DE,HL
        LD      HL,(PATPNT)
JUMP02: OUT     (C),E
        LD      A,36+128
        OUT     (C),A
        OUT     (C),D
        LD      A,38+128
        OUT     (C),A
        OUTI
        LD      A,44+128
        OUT     (C),A
        LD     A,&B01010000
        OUT     (C),A
        LD      A,46+128
        OUT     (C),A
        XOR     A
        INC     E
        INC     E
        ADC     A,D
        LD      D,A
        CALL    WAITDP
        DJNZ    JUMP02
        LD      A,(POINTC)
        INC     A
        CP      NRPATR
        JR      NZ,JUMP03
        XOR     A
        LD      HL,PATTRN
JUMP03: LD      (PATPNT),HL
        LD      (POINTC),A
;set VDP to
;original screen mode
        LD      A,(VDP0AD)
        OUT     (C),A
        LD      A,0+128
        OUT     (C),A
        LD   A,(VDP0AD+1)
        OUT     (C),A
        LD      A,1+128
        OUT     (C),A
        LD   A,(VDP0AD+7)
        OUT     (C),A
        LD      A,7+128
        OUT     (C),A
        XOR     A
        OUT     (C),A
        LD      A,15+128
        OUT     (C),A
        JP      O_TIMI
;wait until VDP is ready
;executing a command
WAITDP: PUSH    BC
        LD      A,(PORT1R)
        LD      C,A
JUMP04: IN      A,(C)
        RRA
        JR      C,JUMP04
        POP     BC
        RET

;data-field
PATPNT: DEFW    PATTRN
POINTC: DEFB    0
O_TIMI: DEFB    0,0,0,0,0
N_TIMI: JP      CODE
        RET
        RET

PORT1R: DEFB    0
PORT1W: DEFB    0
PORT3W: DEFB    0
COUNTR: DEFB    1
PATTRN: DEFB &H90, &H48,
&H24, &H90, &H90, &H24,
&H48, &H90, &H48, &H24,
&H90, &H48, &H48, &h90,
&H24, &H48, &H24, &H90,
&H48, &H24, &H24, &H48,
&H90, &H24
```

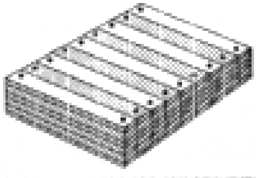
MiLC

MiLC is goed voor elk. Of dat ook voor u zo is kunt u hier uitzoeken...

Als programmeur zijnde zult u waarschijnlijk wel beschikken over een uitgebreide verza-meling documentatie over vele ascpeten van de MSX. Maar als je dan eens een keer wat nodig hebt dan kan je het natuurlijk niet vinden. Na vele uren bladeren vindt je dan eindelijk de bewuste tekst. Om dit soort situaties te vermijden is er binnenkort Milc. Milc is een grote verzameling informatie op MSX gebied. Deze informatie is gebundeld in zogenaamde Milc-files. Deze Milc-files kunnen met een speciaal leesprogramma bekeken worden. Elk van de Milc-files gaat over een bepaald onderwerp. Zo zijn er Milc-files over BASIC, de BIOS, de VDP enz. In een Milc-file zitten diverse teksten over dit onderwerp. De Milc-files kunnen, zoals ge-zegd, worden gelezen met een speciaal programma, de Reader. Om zelf ook informa-tie toe te kunnen voegen wordt ook een Make programma meegeleverd.

Als u zelf Milc-files heeft gemaakt dan kunt u deze informatie uploaden naar uw favoriete BBS. Als u uw informatie opgenomen wil zien in de officiele versie van Milc dan kunt u uw informatie uploaden naar het distributie BBS, dat is InfoDrome BBS (01859-13957). Op dit BBS zullen ook de nieuwste MiLC-files te vinden zijn.

MILC en de bijbehorende MILC-files zijn Public Domain en mogen dus vrij gekopieerd en verspreid worden zonder hiervoor te betalen. Het is toegestaan om zelfgemaakte MILC-files commercieel te verkopen.



MILC wordt op de MSX-beurs in Zandvoort gereleased en is als PD bij MCD te koop.